Project Management System

Introduction

The Builder Portfolio Management System is a console-based Java application intended to help the Builder Association manage their construction projects more efficiently and automatically. It replaces the traditional spreadsheet-based management system.The Builder Portfolio Management System supports various user roles, including Builder, Project Manager, and Client, with their respective permissions and responsibilities. It supports the creation of projects, managing tasks, assigning tasks, and updating the status of the projects using background processing. It ensures the efficiency and correctness of the application by using validation, access control, and modular programming (DAO, Service, View).

Problem Statement

The Builders' Association aims to transition from manual, spreadsheet based project record management to a robust, real-time, console-based java application. The current manual process leads to data inconsistency, loss, errors, and inefficient communication with clients and stakeholders. To address these issues, you are tasked with designing and developing a Builder Portfolio Management System that supports real-time data handling, multiple user roles, and reliable project management functionalities.

User Stories

As a user, i can register and securely login to the application

As a Client, I want to create a new project by providing project name , So that project manager can initiate work

As a Builder,I want to work on tasks within a project,So that project can be broken down into manageable units.

As a Builder,I want to update my progress

As a Project Manager,I want to view available tasks,So that I can manage task assignments.

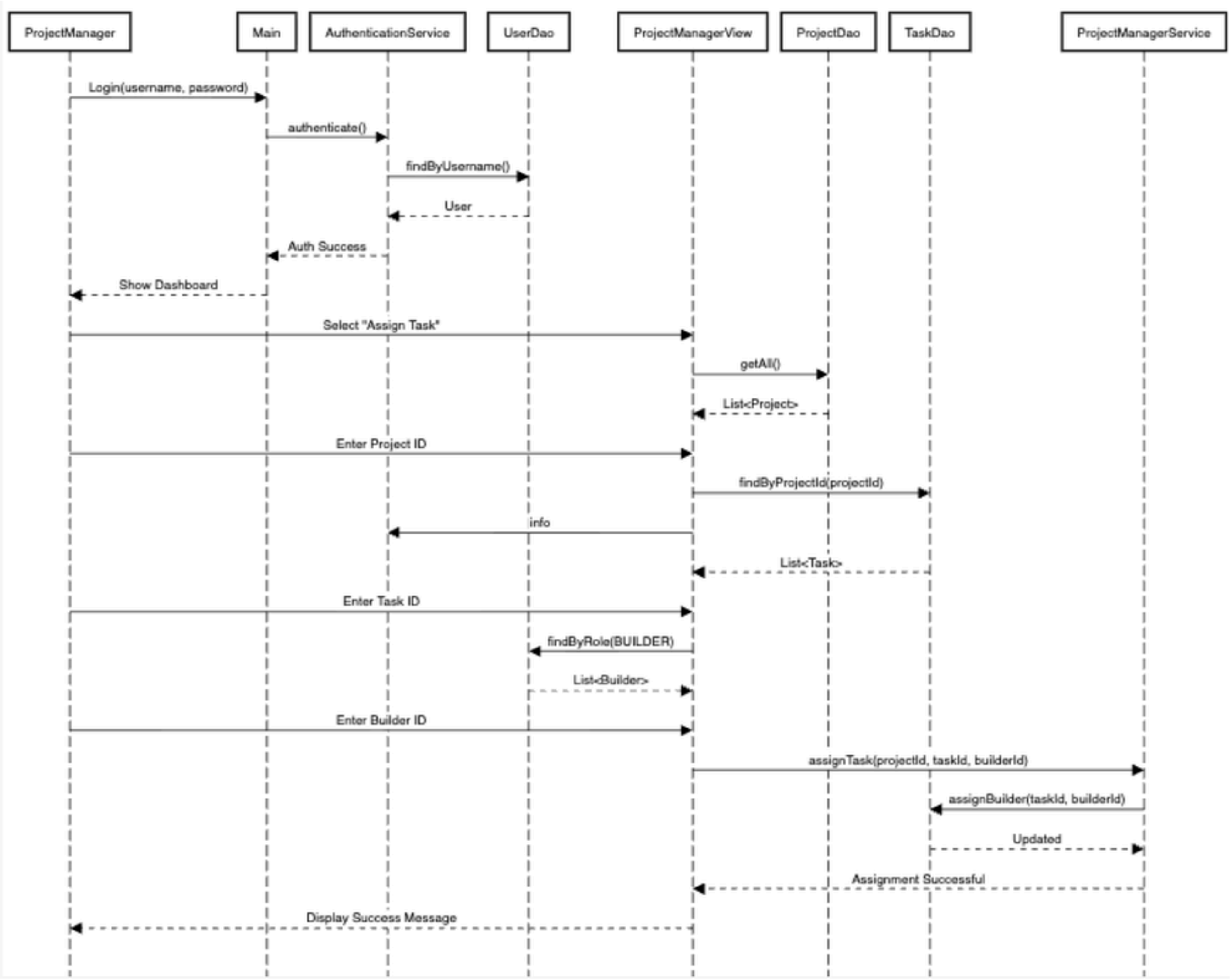
As a Project Manager,I want to update or override task status if needed, So that project tracking remains accurate.

As a Project Manager,I want to create and assign task to the builders.

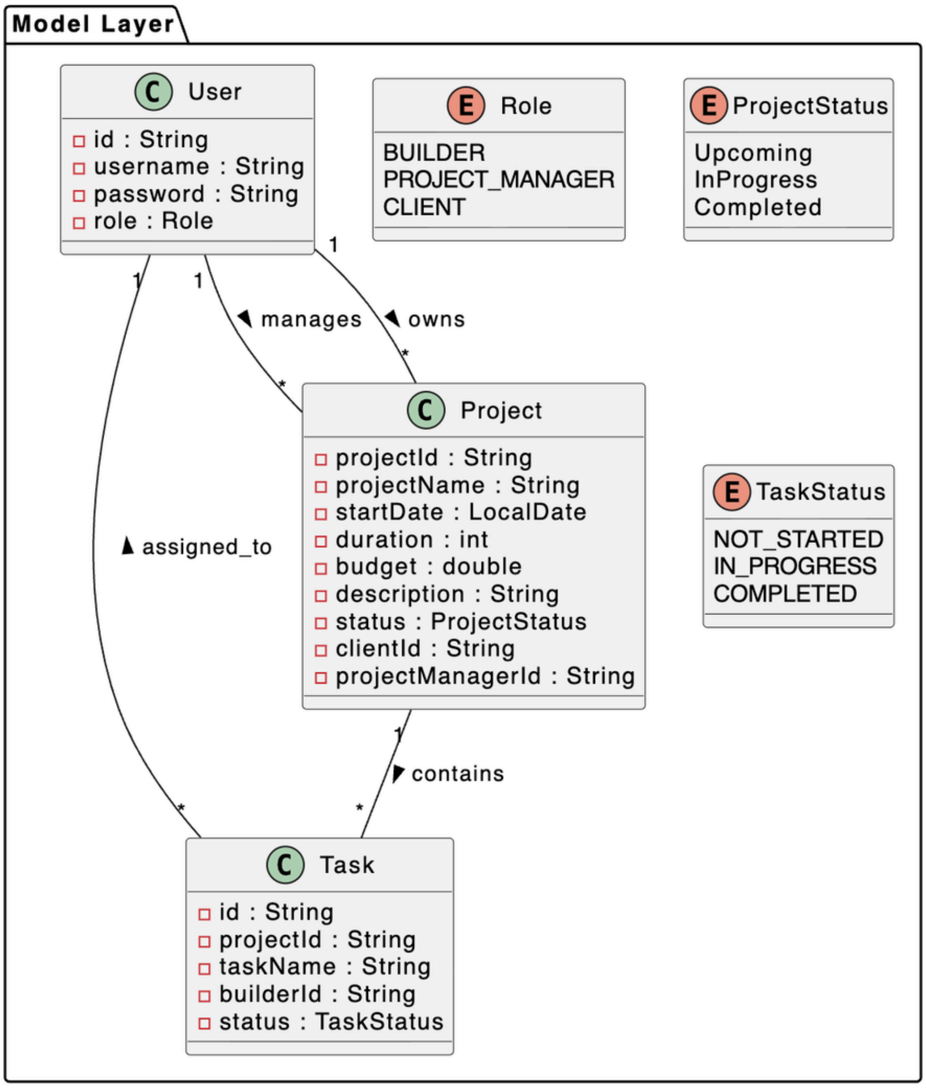
As a Project Manager,I want to track list of all the project i am assigned to.

As a Client,I want to view projects associated with me, So that I can track progress.

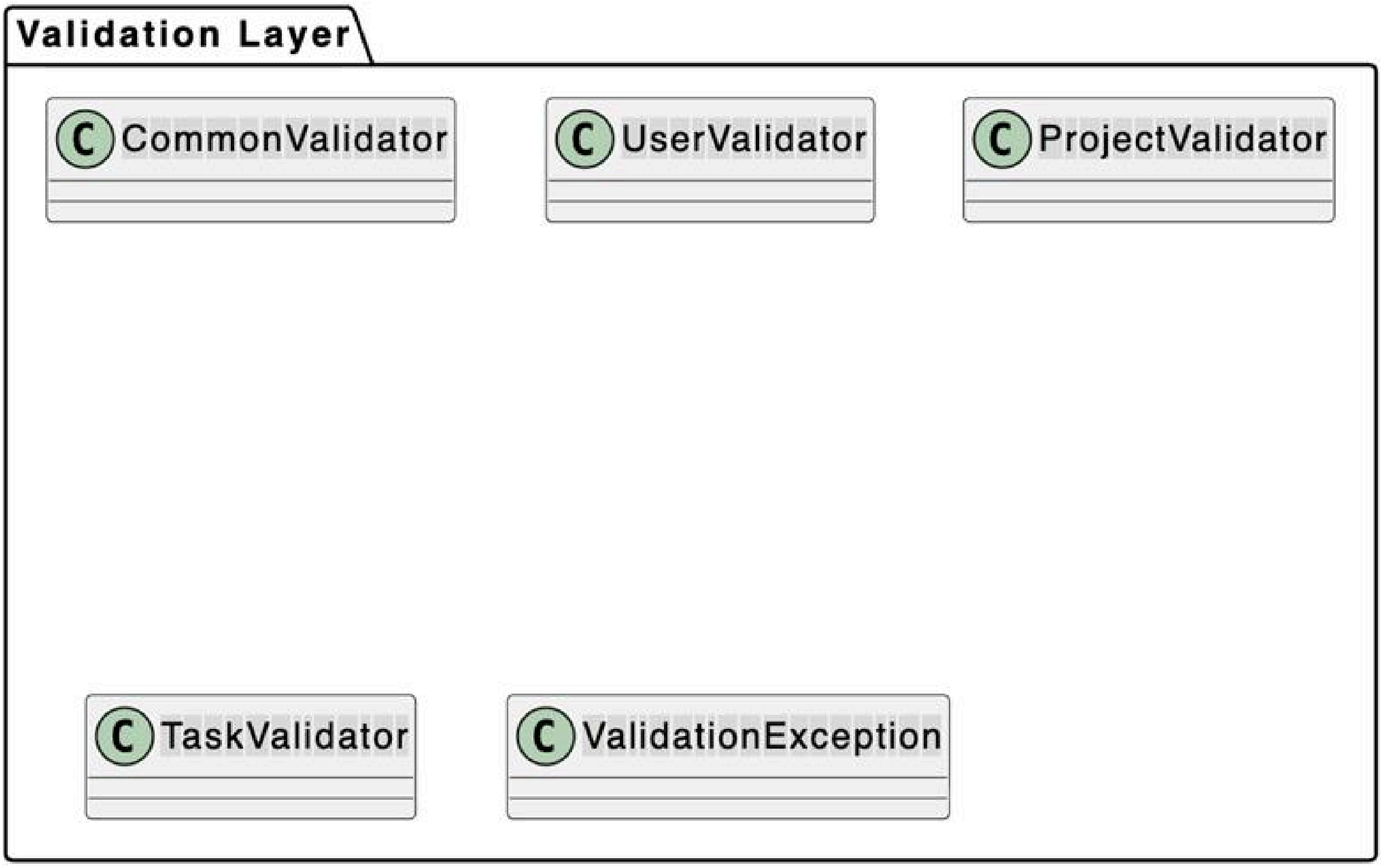
Sequence Diagram

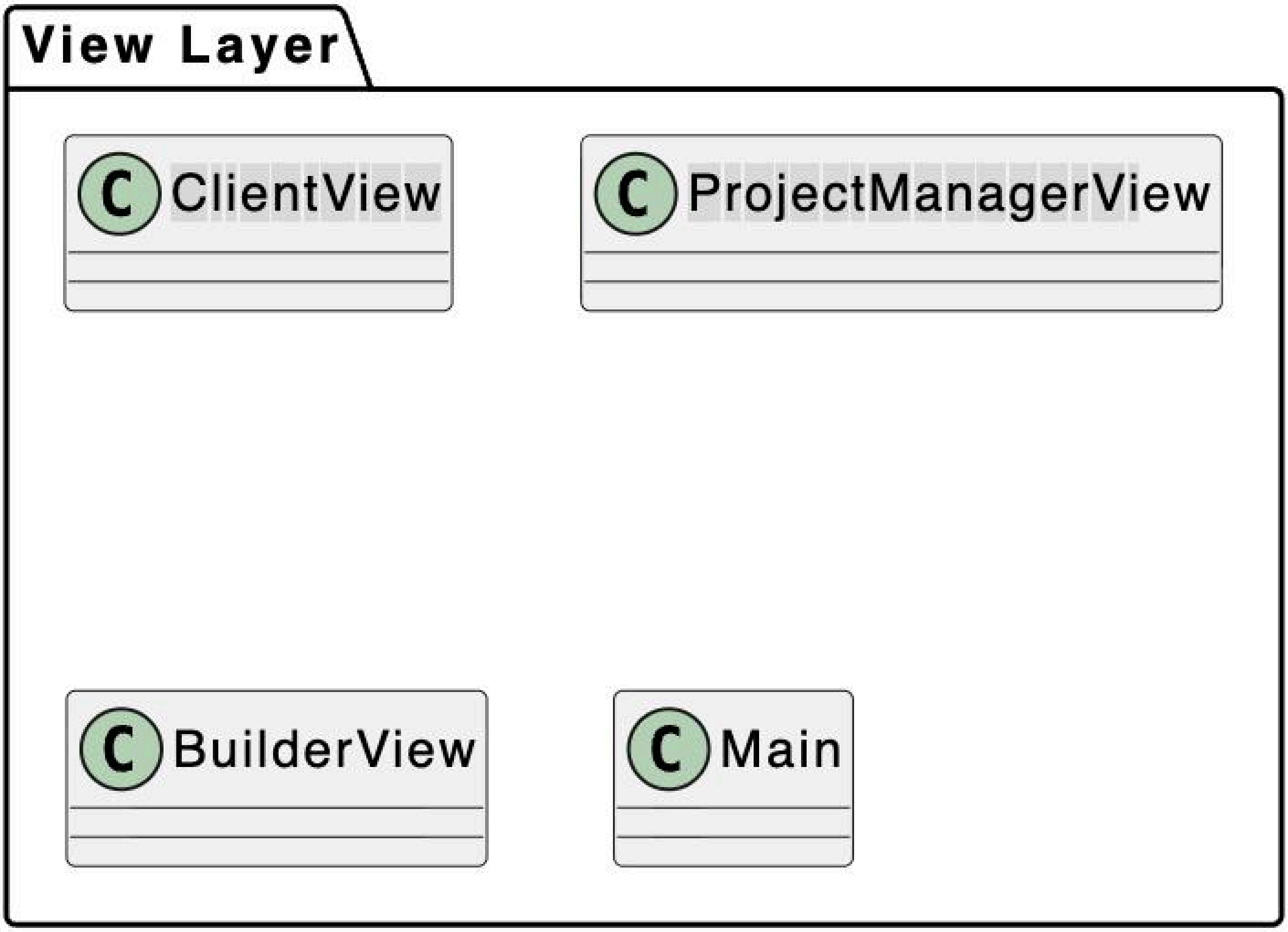


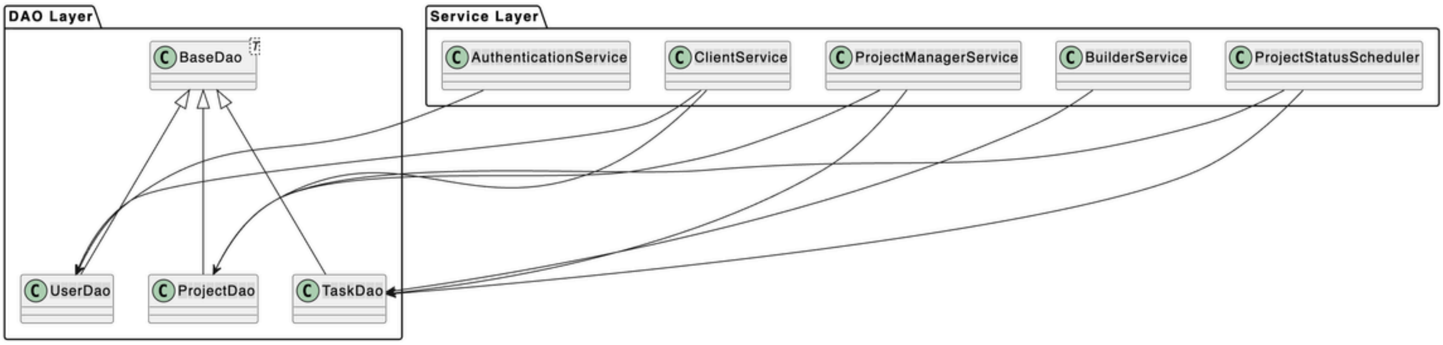
Class Diagram



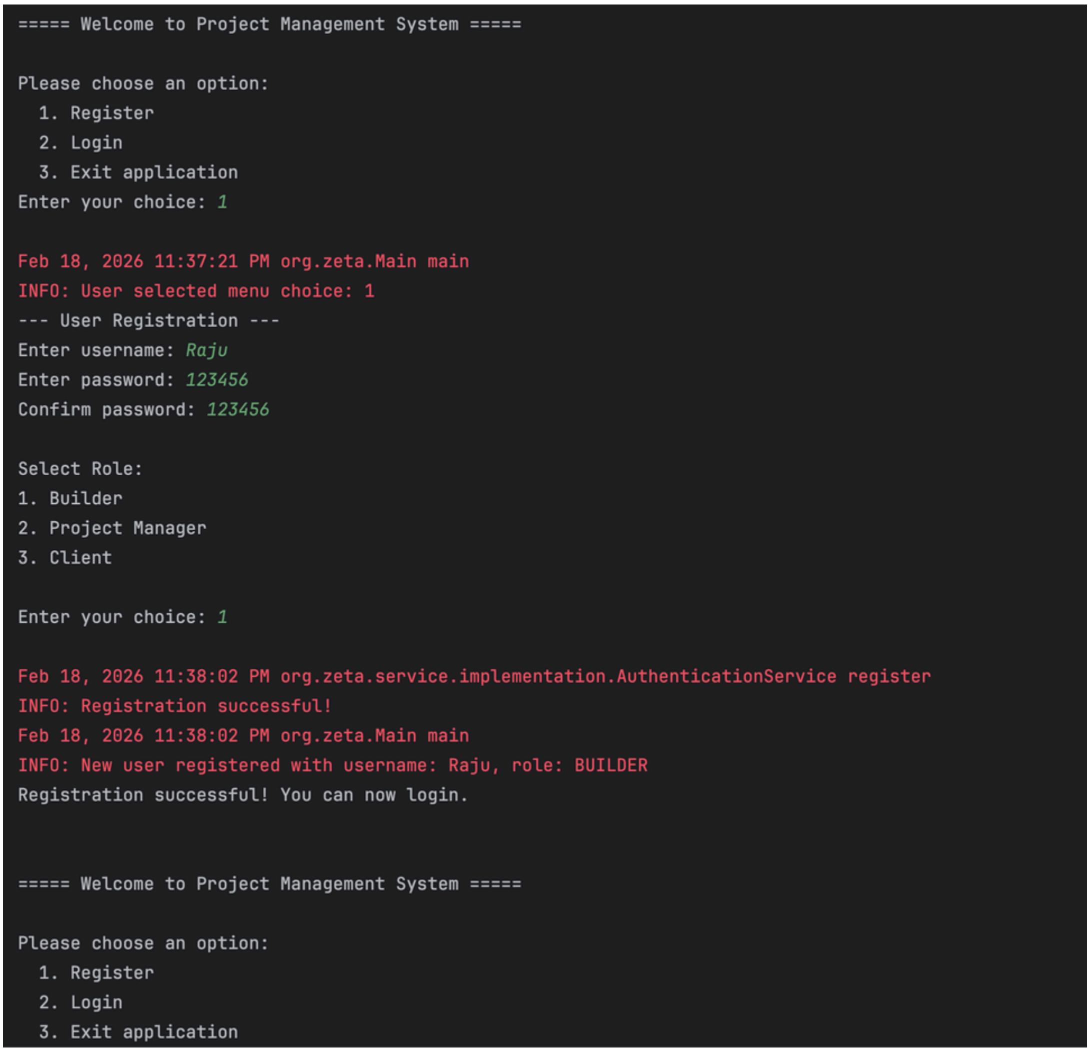
The class diagram essentially describes the fundamental structure of the Builder Portfolio Management System. It describes the interaction of users, projects, and tasks in the system. In the class diagram, the User class describes the participants in the system, including the Builder, Project Manager, and Client. The Project class describes the data associated with a project, including the schedule, budget, status, and ownership by the client and management by the project manager. Finally, the Task class describes the tasks performed by the builder in a project, including the status of the tasks. These classes and their relationships essentially describe a hierarchical workflow in the system, where the client initiates the project, the project manager monitors and manages the project, and the builder performs the tasks in the project.


Validation Layer
CommonValidator
UserValidator
ProjectValidator
TaskValidator
ValidationException


View Layer
ClientView
ProjectManagerView
BuilderView
Main

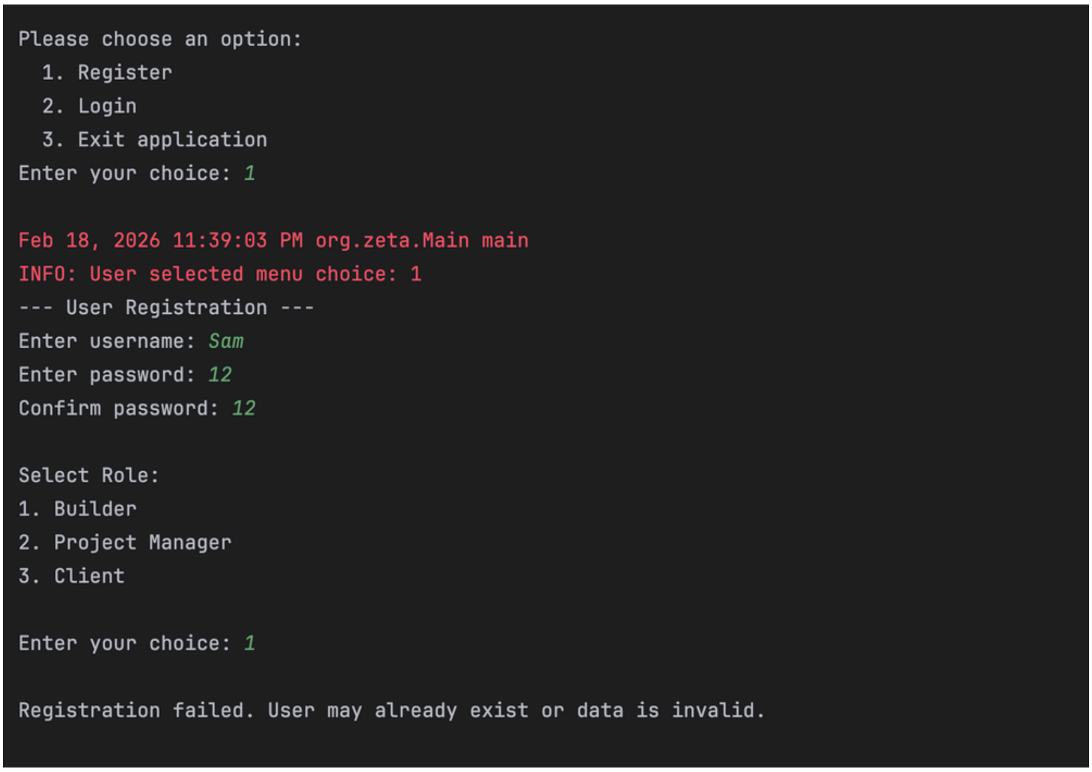


Implementation



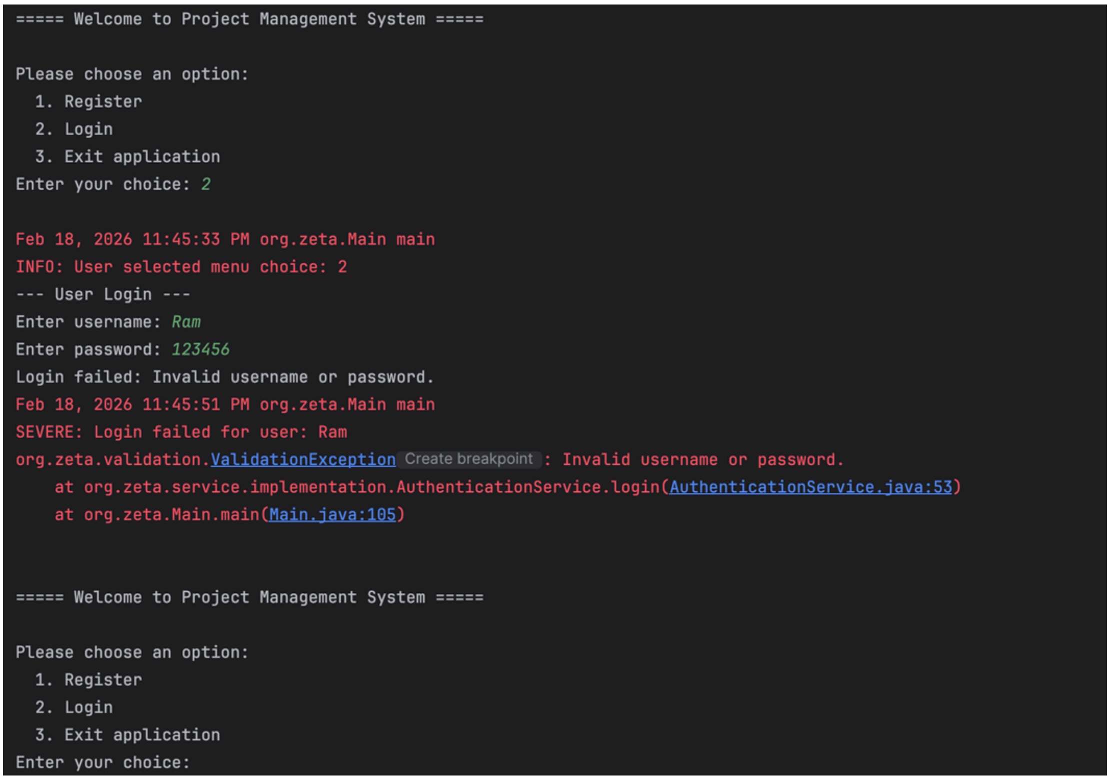
```
===== Welcome to Project Management System =====

Please choose an option:
  1. Register
  2. Login
  3. Exit application
Enter your choice: 1

Feb 18, 2026 11:37:21 PM org.zeta.Main main
INFO: User selected menu choice: 1
--- User Registration ---
Enter username: Raju
Enter password: 123456
Confirm password: 123456

Select Role:
1. Builder
2. Project Manager
3. Client

Enter your choice: 1

Feb 18, 2026 11:38:02 PM org.zeta.service.implementation.AuthenticationService register
INFO: Registration successful!
Feb 18, 2026 11:38:02 PM org.zeta.Main main
INFO: New user registered with username: Raju, role: BUILDER
Registration successful! You can now login.


===== Welcome to Project Management System =====

Please choose an option:
  1. Register
  2. Login
  3. Exit application
```

Registration Successful



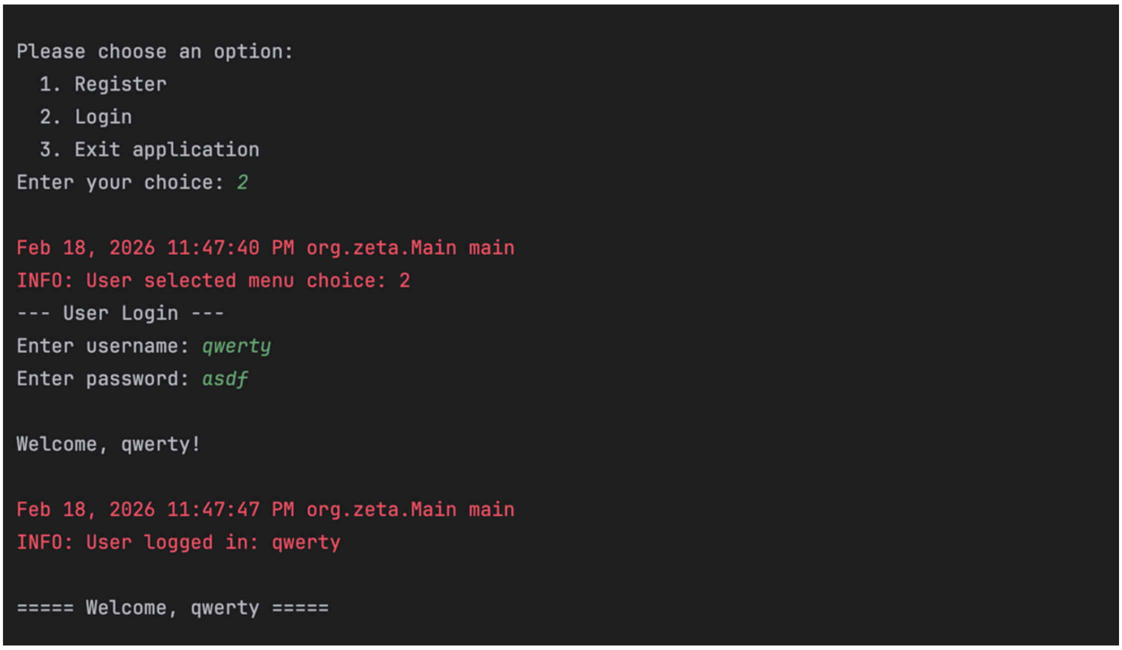
```
Please choose an option:
  1. Register
  2. Login
  3. Exit application
Enter your choice: 1

Feb 18, 2026 11:39:03 PM org.zeta.Main main
INFO: User selected menu choice: 1
--- User Registration ---
Enter username: Sam
Enter password: 12
Confirm password: 12

Select Role:
1. Builder
2. Project Manager
3. Client

Enter your choice: 1

Registration failed. User may already exist or data is invalid.
```

Registration failed,Already exiting user cannot register



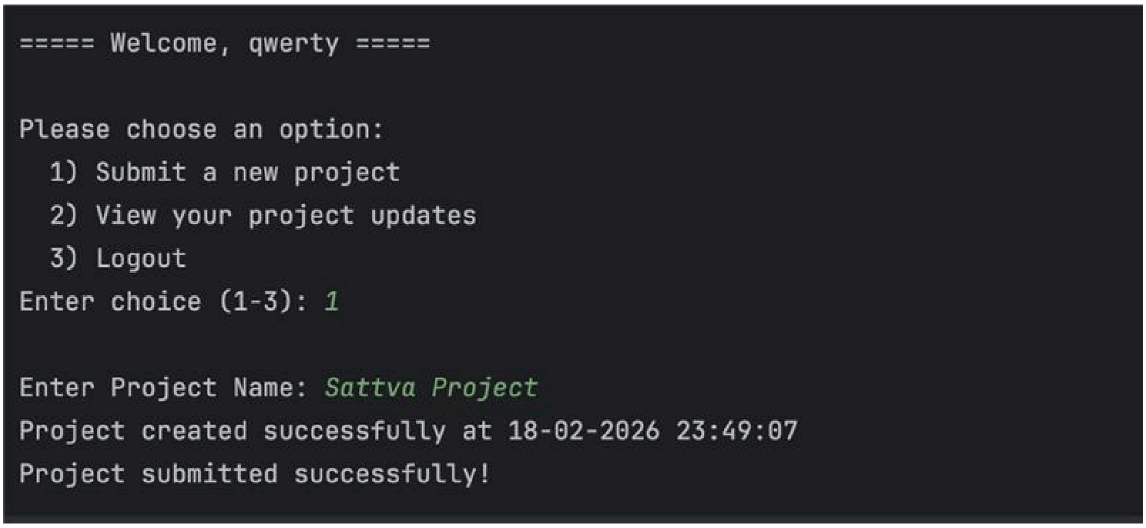
```
===== Welcome to Project Management System =====

Please choose an option:
  1. Register
  2. Login
  3. Exit application
Enter your choice: 2

Feb 18, 2026 11:45:33 PM org.zeta.Main main
INFO: User selected menu choice: 2
--- User Login ---
Enter username: Ram
Enter password: 123456
Login failed: Invalid username or password.
Feb 18, 2026 11:45:51 PM org.zeta.Main main
SEVERE: Login failed for user: Ram
org.zeta.validation.ValidationException Create breakpoint : Invalid username or password.
    at org.zeta.service.implementation.AuthenticationService.login(AuthenticationService.java:53)
    at org.zeta.Main.main(Main.java:105)


===== Welcome to Project Management System =====

Please choose an option:
  1. Register
  2. Login
  3. Exit application
Enter your choice:
```

Login failed, User does not exits



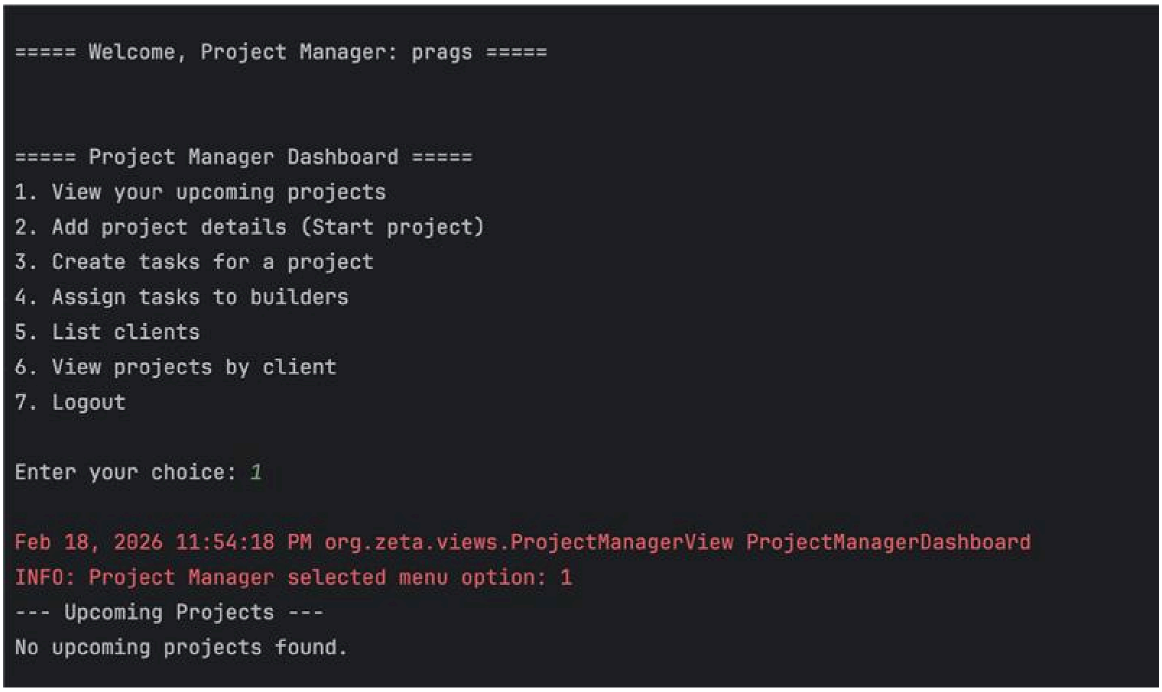
```
Please choose an option:
  1. Register
  2. Login
  3. Exit application
Enter your choice: 2

Feb 18, 2026 11:47:40 PM org.zeta.Main main
INFO: User selected menu choice: 2
--- User Login ---
Enter username: qwerty
Enter password: asdf

Welcome, qwerty!

Feb 18, 2026 11:47:47 PM org.zeta.Main main
INFO: User logged in: qwerty

===== Welcome, qwerty =====
```

Login Successful



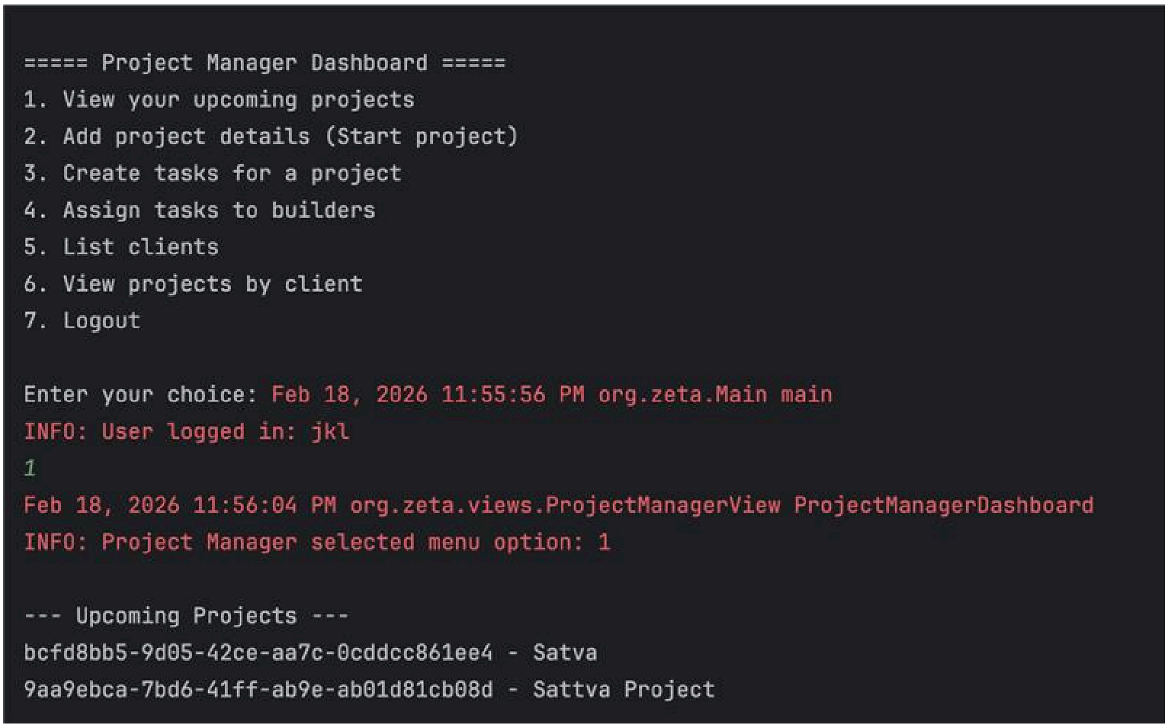
```
===== Welcome, qwerty =====

Please choose an option:
  1) Submit a new project
  2) View your project updates
  3) Logout
Enter choice (1-3): 1

Enter Project Name: Sattva Project
Project created successfully at 18-02-2026 23:49:07
Project submitted successfully!
```

Client is able to submit Projects



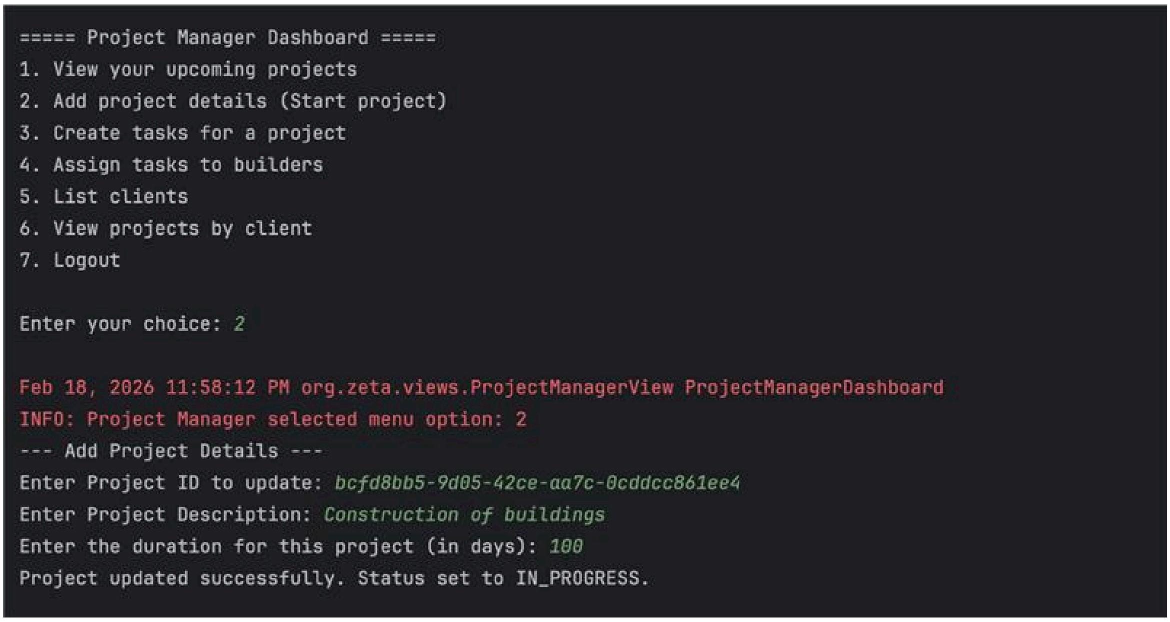
```
===== Welcome, Project Manager: prags =====


===== Project Manager Dashboard =====
1. View your upcoming projects
2. Add project details (Start project)
3. Create tasks for a project
4. Assign tasks to builders
5. List clients
6. View projects by client
7. Logout

Enter your choice: 1

Feb 18, 2026 11:54:18 PM org.zeta.views.ProjectManagerView ProjectManagerDashboard
INFO: Project Manager selected menu option: 1
--- Upcoming Projects ---
No upcoming projects found.
```

ProjectManager Dashboard



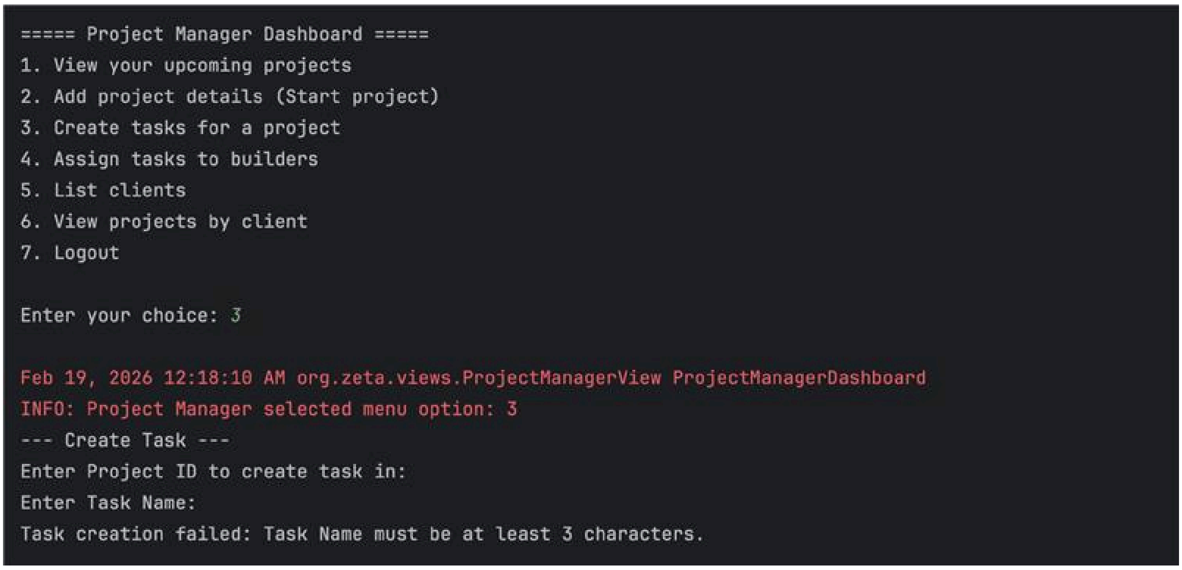
```
===== Project Manager Dashboard =====
1. View your upcoming projects
2. Add project details (Start project)
3. Create tasks for a project
4. Assign tasks to builders
5. List clients
6. View projects by client
7. Logout

Enter your choice: Feb 18, 2026 11:55:56 PM org.zeta.Main main
INFO: User logged in: jkl
1
Feb 18, 2026 11:56:04 PM org.zeta.views.ProjectManagerView ProjectManagerDashboard
INFO: Project Manager selected menu option: 1

--- Upcoming Projects ---
bcfd8bb5-9d05-42ce-aa7c-0cddcc861ee4 - Satva
9aa9ebca-7bd6-41ff-ab9e-ab01d81cb08d - Sattva Project
```

ProjectManager is able to view Upcoming Projects



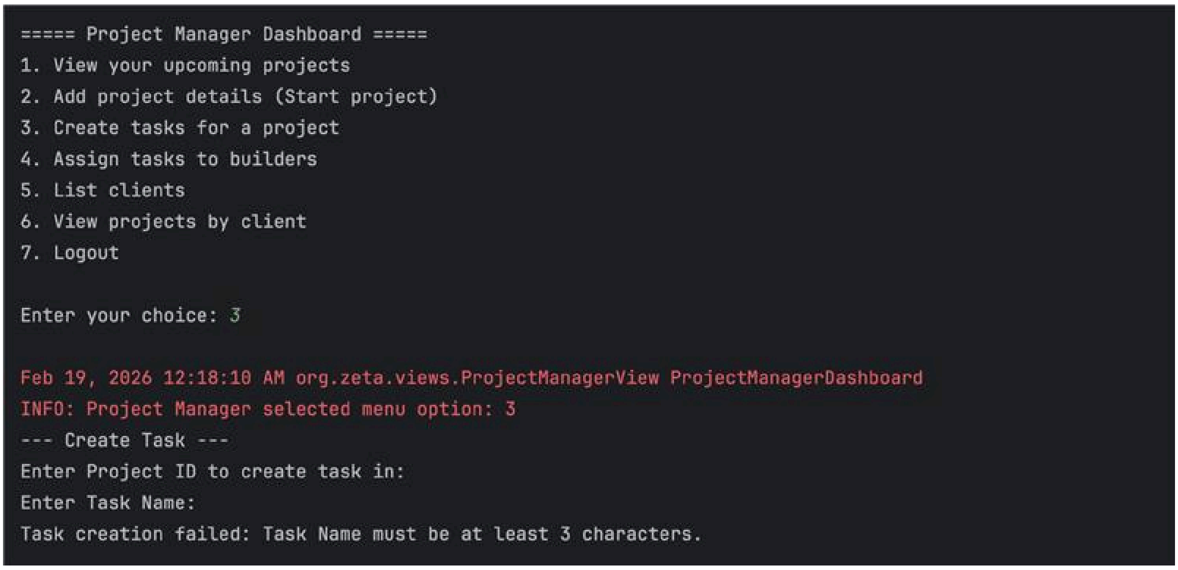
```
===== Project Manager Dashboard =====
1. View your upcoming projects
2. Add project details (Start project)
3. Create tasks for a project
4. Assign tasks to builders
5. List clients
6. View projects by client
7. Logout

Enter your choice: 2

Feb 18, 2026 11:58:12 PM org.zeta.views.ProjectManagerView ProjectManagerDashboard
INFO: Project Manager selected menu option: 2
--- Add Project Details ---
Enter Project ID to update: bcfd8bb5-9d05-42ce-aa7c-0cddcc861ee4
Enter Project Description: Construction of buildings
Enter the duration for this project (in days): 100
Project updated successfully. Status set to IN_PROGRESS.
```

ProjectManager can add details to Upcoming project and set it in IN_PROGRESS



```
===== Project Manager Dashboard =====
1. View your upcoming projects
2. Add project details (Start project)
3. Create tasks for a project
4. Assign tasks to builders
5. List clients
6. View projects by client
7. Logout

Enter your choice: 3

Feb 19, 2026 12:18:10 AM org.zeta.views.ProjectManagerView ProjectManagerDashboard
INFO: Project Manager selected menu option: 3
--- Create Task ---
Enter Project ID to create task in:
Enter Task Name:
Task creation failed: Task Name must be at least 3 characters.
```

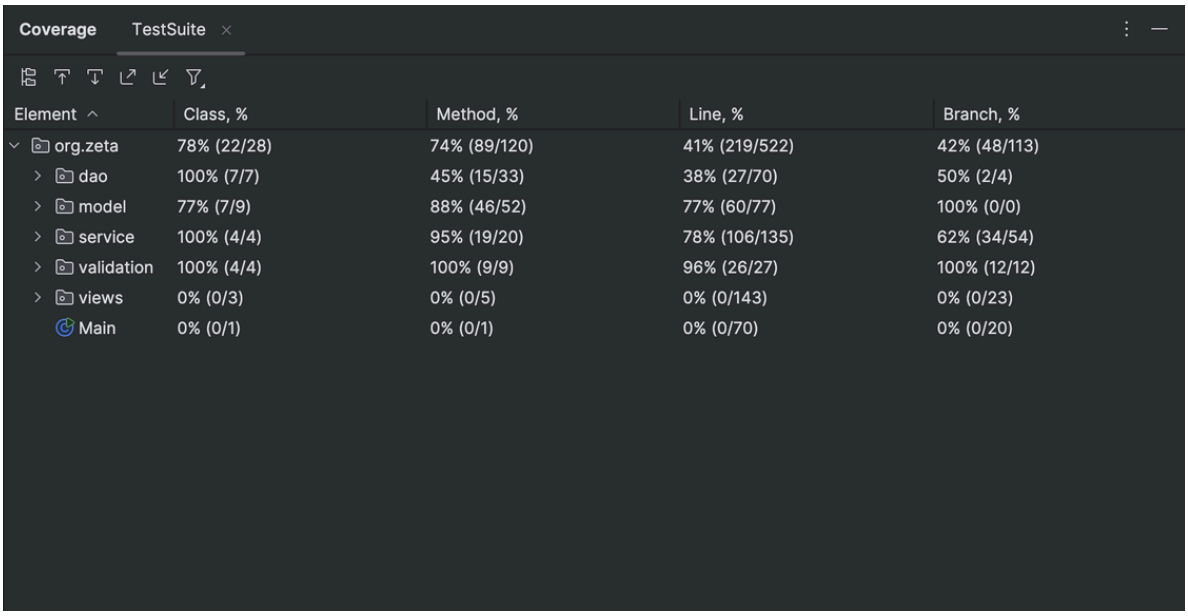
ProjectManager can create Tasks in IN_PROGRESS projects



```
===== Project Manager Dashboard =====
1. View your upcoming projects
2. Add project details (Start project)
3. Create tasks for a project
4. Assign tasks to builders
5. List clients
6. View projects by client
7. Logout

Enter your choice: 3

Feb 19, 2026 12:18:10 AM org.zeta.views.ProjectManagerView ProjectManagerDashboard
INFO: Project Manager selected menu option: 3
--- Create Task ---
Enter Project ID to create task in:
Enter Task Name:
Task creation failed: Task Name must be at least 3 characters.
```

Validation checks while creating tasks

Code Coverage



Element	Class, %	Method, %	Line, %	Branch, %
org.zeta	78% (22/28)	74% (89/120)	41% (219/522)	42% (48/113)
dao	100% (7/7)	45% (15/33)	38% (27/70)	50% (2/4)
model	77% (7/9)	88% (46/52)	77% (60/77)	100% (0/0)
service	100% (4/4)	95% (19/20)	78% (106/135)	62% (34/54)
validation	100% (4/4)	100% (9/9)	96% (26/27)	100% (12/12)
views	0% (0/3)	0% (0/5)	0% (0/143)	0% (0/23)
Main	0% (0/1)	0% (0/1)	0% (0/70)	0% (0/20)

Tech Stack

Category	Technology	Purpose
Backend	Java (JDK 8 or above)	Core application logic
Programming Concepts	OOP, Design Patterns (DAO Pattern, Service Layer)	Structured and maintainable code architecture
Data Storage	JSON File	Data persistence instead of relational database
Database Connectivity	File I/O (Jackson/Gson if used)	Reading and writing data to JSON files
Build Tool	Maven	Dependency management & build automation
Testing	JUnit 5	Unit testing
Version Control	Git	Source code version control
Repository Hosting	GitHub	Remote repository management
Architecture	Layered Architecture	Separation of concerns
Layers	Model Layer	Entity/data objects
	DAO Layer	Data access from JSON files
	Service Layer	Business logic
	Console-based UI Layer	User interaction via terminal

Conclusion

The Builder Portfolio Management System is a well-structured, console-based Java program developed with the principles of layered architecture. The project follows the separation of concerns principle by implementing the Model, DAO, and Service layers. Object-oriented programming concepts and design patterns have also been applied to ensure better code reusability and organization.If database integration, unit testing, and version control are implemented correctly in the project, it will provide an efficient solution for builder portfolio management. The project reflects an in-depth understanding of basic Java programming concepts, database connectivity, software design principles, and clean architecture practices.